MOTION SEGMENTATION ON THE TMS320C80 MULTIMEDIA VIDEO PROCESSOR

R. J. Fergusson, J. J. Soraghan

Signal Processing Division, University of Strathclyde
George Street, Glasgow G1 1XW, Scotland UK
e-mail: robert@spd.eee.strath.ac.uk, jjs@spd.eee.strath.ac.uk

ABSTRACT

In MPEG-4 that uses Video Object Planes (VOP's), systems to efficiently extract objects are required. Based on research into the Human Visual System (HVS), this paper presents an Object Extraction System that uses Motion Segmentation and Image Segmentation in parallel, and then combines them to efficiently extract objects using Fuzzy Reasoning. The most computationally demanding part of this system, the Motion Segmentation (MS), is implemented on the TMS320C80 DSP MVP. The Motion Estimation part of the MS algorithm and its fast implementation on C80 architecture are also described.

1. INTRODUCTION

Since the mid-eighties, there has been an increasing interest in object based techniques, due to its potential to be more perceptually pleasing, and having greater flexibility than block or pixels based techniques such as H.261 and H.263. MPEG-4 [7] has recently been defined, and is an object based technique. Exactly how objects are extracted was not defined in the standard. Techniques to extract objects, either by using Motion Segmentation [8] or Image Segmentation [5] alone tend to be computationally demanding and complex. This paper presents a system that uses relatively simple Motion Segmentation and Image Segmentation to extract objects. Motion and Image Segmentation are carried out in parallel, with their outputs being combined to use their respective advantages to extract objects. The Motion Segmentation is the most computationally demanding part, and a fast implementation on the TMS320C80 [9] DSP Multimedia Video Processor (MVP) is described. This work extends that presented at VLBV-97 [2], and IASTED-98 [3].

2. PROPERTIES OF THE HVS

The HVS based object extraction system is based on research into the HVS, that has revealed that there are two main pathways called the M and P pathways [1], [6] that are fundamental to our ability to code the visual scene. The M pathway is dedicated to motion and depth perception, where as the P pathway is dedicated to form and color perception. The M pathway spans from the M ganglion cells in the Retina to the Parietal Cortex. The P pathway spans from the P ganglion cells in the Retina to the Temporal Cortex. The earlier stages in these pathways are of most interest to us, from the Retina to multipurpose visual areas V1 and V2, to V3 specialized for form, and V5 specialized for motion. The Retina uses a very high sampling density at the center of vision for coding differences in luminosity. In the M pathway V1 and V2 extract the local motion, and V5 uses this local motion to form coherent regions of motion, thus extracting global motion. In the P pathway V1 and V2 segment the visual scene into boundaries, and V3 then combines them to form regions. Both pathways interact with each other to associate regions of motion with segmented regions, i.e. to associate different attributes to the same object. This interaction also allows the enhancement of form and motion boundaries.

3. OBJECT EXTRACTION SYSTEM

The HVS based object extraction system presented consists of motion and image segmentation in parallel, and their combination, as illustrated in figure 1.

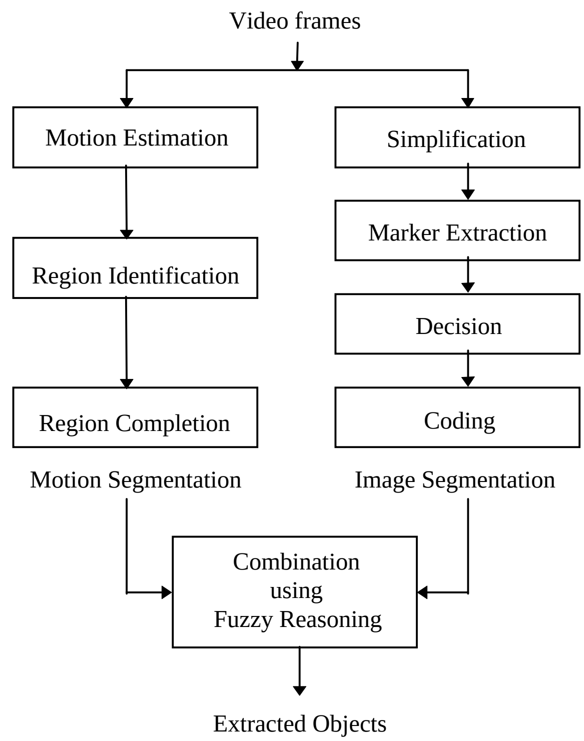

Figure 1. Object Extraction System

3.1 Motion Segmentation

The Motion Segmentation pathway consists of *Motion Estimation*, *Region Identification*, and *Region Completion*. *Motion Estimation* uses full search block matching on 8 x 8 overlapping blocks, using MAE. Maximum overlapping blocks are used (i.e. blocks overlapping to within one pixel of the previous block), which significantly improves the coherence of the motion fields produced. The images used are Interpolated to obtain half pel accuracy. The Interpolated images are then *Differenced* so to use the contrast values of the images rather than the intensity values. Differencing the images reduces the probability of erroneous motion vectors caused by intensity variations between frames. Maximum overlap and Differencing are based on the properties of the Retina in the eye. *Region Identification* is used to find the most important motion vectors, which tend to be correlated into regions that represent objects. Motion vectors that are uncorrelated with their surround are taken as erroneous motion vectors and are reduced to zero. Correlated motion vectors are found by searching the surrounding motion vectors, if there is more than one motion vector that is similar then the motion vector is left unaltered. If there is one or less similar motion vectors, then the motion vector is reduced to zero. However, motion vectors that have been reduced to zero can leave holes in regions of motion. *Region Completion* is used to fill these holes, which increases the coherency of the motion fields. A correlation mask is used to analyze the surrounding motion vectors to find the highest correlating motion vector to fill the hole. Figure 3 shows the output of the Motion Segmentation pathway of the original shown in figure 2.

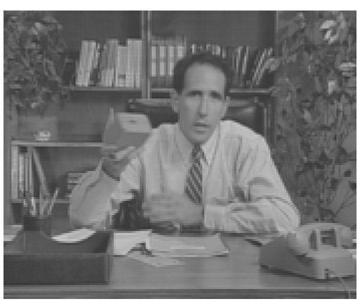

Figure 2. Original Salesman

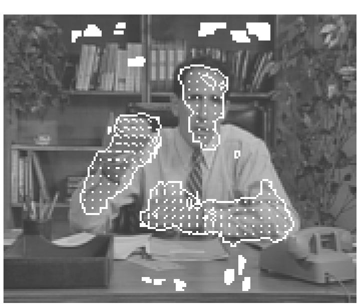

Figure 3. Motion Segmentation

3.2 Image Segmentation

The Image Segmentation pathway is based on a Multiscale Morphological Image Segmentation technique [4]. The method uses a hierarchical structure approach, where an iterative process is carried out on a segmentation stage. The segmentation stage consists of four parts. *Simplification* which removes information from the image to make it easier to segment. *Marker Extraction* is used to identify homogeneous regions, and a *Decision* step is used to precisely locate the contours of the regions detected by the Marker Extraction step. Finally, a *Coding* step is used to code all the information about regions that have not been properly segmented by the Decision step. The segmentation step is carried out five times, with a decreasing structuring element size that reveals an increasing number of regions after each step. The number of region revealed depends on the size and content of the image. Figure 4 shows the output of the Image Segmentation pathway.

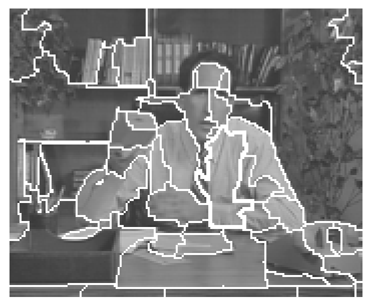

Figure 4. Image Segmentation

3.3 Combination

The Combination stage combines the outputs of the Motion and Image Segmentation pathways, using the advantages of each method to extract objects. The Motion Segmentation has the advantage that it locates objects, even though the boundaries may be coarse. The Image Segmentation has the advantage that it precisely locates region boundaries, but these regions do not necessary represent objects. *Fuzzy Reasoning* and the Difference image is used to choose between the Motion and Image Segmentation boundaries. The Difference image is used to associate contrast weightings with the Motion and Image Segmentation boundaries and the space in between, so that the Fuzzy Reasoning can select the boundary that best represents the object. Figure 5 shows the combination of the Motion and Image Segmentation outputs. Figure 6 shows the extracted objects.

Figure 5. Combination

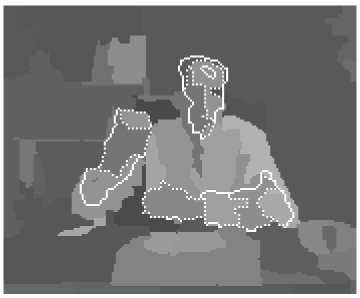

Figure 6. Extracted objects

4. C80 IMPLEMENTATION OF MOTION SEGMENTATION

The Motion Segmentation pathway requires considerably more computation than the Image Segmentation pathway or the Combination stage. Hence a TMS320C80 DSP platform is being used to address the computational demands of this pathway. Since the MVP was specifically designed for multimedia applications, the Motion Segmentation algorithm maps very efficiently onto its architecture. There are five main parts to the Motion Segmentation algorithm:

- Motion Estimation (ME)
- Rectangular to Polar Conversion (R⟶P)
- Region Identification (RI)
- Polar to Rectangular Conversion (P⟶R)
- Region Completion (RC)

The ME part generates the motion vectors, and the RI and RC parts process them. The motion vectors generated by the ME part are represented in Rectangular coordinates. For the RI part, they have to be converted to Polar coordinates, since the magnitude and phase of the motion vectors are needed to determine whether motion vectors are correlated with their surround. The motion vectors are then converted back to Rectangular coordinates for the RC part, and for displaying the motion field if necessary. The parts must be processed in order, so there are two possible ways to map them onto the C80, either in the form of a *Pipeline*, or an *Array*. The main drawback of the Pipeline implementation is that there are five parts, and that each part has different computational demands. Thus the PP's would not be used very efficiently, as they would be idle for a large proportion of the time. However, an Array implementation does not suffer from these problems, as each part uses all four PP's, with each working on a quarter of the data set. Figure 7 shows a block diagram of the implementation of Motion Segmentation on the C80.

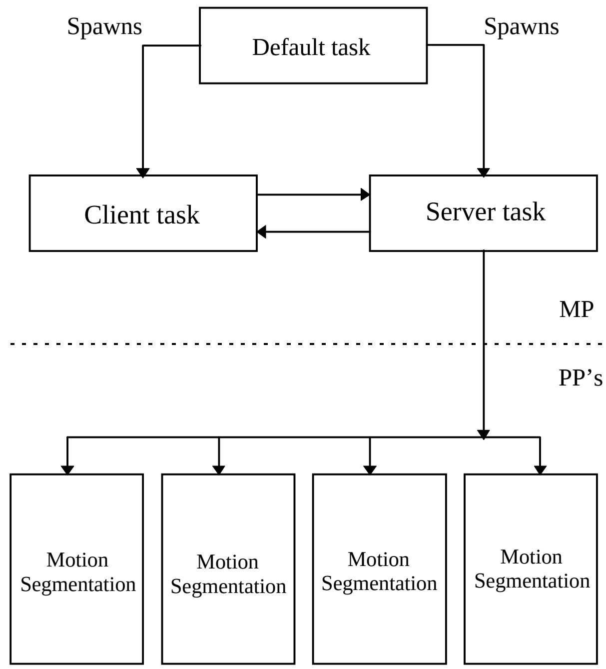

Figure 7. TMS320C80 Implementation

4.1 Motion Estimation

The Motion Estimation (ME) part of the Motion Segmentation algorithm is being implemented on the C80. The ME part comprises seven main parts:

- Set up of ME parameters (MP)
- Interpolation (MP)
- Differencing (MP)
- Send ME parameters to PP's
- Set up Packet Transfers (MP)
- Construct Carry Look up table (PP)
- Motion Estimation (PP)

The first part sets up the PP's for partitioning the image so that each PP processes a quarter of the image, and also for setting up the images. The next two parts carry out the Interpolate and Difference operations on the images. The fourth part sends the parameters to the PP's, for ME processing. The parameters are the block size, search area size, number of blocks, the overlap.

The fifth part sets up Packet Transfers for loading in the image data, and storing the motion vectors. The sixth part generates a carry look up table in the PP's memory to allow four pixels to be processed at once, rather than just one, and to aid in calculation of the MAD (Mean Absolute Difference). The last part carries out the Motion Estimation on a pair of images, with each PP processing a quarter of the image. All code is written in assembly language to use the full processing capabilities of the C80.

5. SUMMARY

This paper has presented an Object Extraction System that uses both Image and Motion Segmentation in parallel, and combines them to efficiently extract objects using Fuzzy Reasoning. A fast implementation of the most computationally demanding part, Motion Segmentation, is implemented on the TMS320C80 MVP.

6. ACKNOWLEDGEMENTS

Sponsored by a bicentennial University of Strathclyde Studentship, GPT Video Systems, and with collaboration from Texas Instruments under their DSP ELITE programme.

7. REFERENCES

[1] Fergusson, R. J., 'Advanced Video Coding: Investigation of Human Visual Neural Characteristics, and Compression Techniques,' Technical Rep., Dep. Eng., Uni. Strathclyde, Feb. 1996.

[2] Fergusson, R. J., and Soraghan, J. J., 'Human Visual System based Video Coding,' Proc. Int. Workshop on VLBV Coding, Sweden, July 1997.

[3] Fergusson, R. J., and Soraghan, J. J., 'Human Visual System based Segmentation,' Proc. IASTED Int. Conf. on Signal Processing and Communications, Canary Islands, Feb. 1998.

[4] Salembier, P., and Pardas, M., 'Hierarchical morphological segmentation for image sequence coding,' IEEE Trans. IP, vol. 3, No. 5, Sept. 1994.

[5] Salembier, P., Marques, F., Pardas, M., Morros, J. R., Corset, I., Jeannin, S., Bouchard, L., Meyer, F., Marcotegui, B., 'Segmentation-Based Video Coding System Allowing the Manipulation of Objects,' IEEE Trans. Circ. Syst. Video Technology, Vol. 7, No. 1, pp. 60-74, February 1997.

[6] Schiller, P. H., 'The central visual system,' Vision Res., Vol. 26, pp. 1351-1386, 1986.

[7] Sikora, T., 'The MPEG-4 Video Standard Verification Model,' IEEE Trans. Circ. Syst. Video Technology, Vol. 7, No. 1, pp. 19-31, February 1997.

[8] Stiller, C., 'Object-Based Estimation of Dense Motion Fields,' IEEE Trans. IP, Vol. 6, No. 2, pp. 234-250, February 1997.

[9] Texas Instruments, TMS320C80 Technical Brief: Media Video Processor, 1994.